Authentication

This page describes the authentication mechanism for bcfg2 clients. All of this functionality is driven from the Metadata/clients.xml file. Each entry in clients.xml can have the following entries. (* means required)

Field	Values	Default	Result
name*	hostname	none	establishes the client node name
profile*	valid profiles	default is established in groups.xml	Sets the client's overall profile group
uuid	string	none	Establishes a per−node name that can be used to bypass dns−based client resolution
password	string	none	Establishes a per−node password that can be used instead of the global password
address	ip address	none	Establishes an extra ip address that resolves to this client
secure	true|false	false	Requires the use of the per−client password for this client
location	fixed|floating	fixed	Requires requests to come from an IP address that matches the client record

Scenarios

1. Cluster nodes that are frequently rebuilt
Default settings work well; machines do not float, and a per−client password is not required.

2. NAT
Build client records in advance with bcfg2−admin, setting a uuid for each new client. Set the address attribute for each to the address of the NAT. Optionally, set a per−client password for each, and set into secure mode. This will require the use of the uuid and password from each client, and will require that they come through the NAT address.

Building bcfg2.conf automatically

This is a TCheetah template that automatically constructs per−client bcfg2.conf from the per−client metadata.

```
[communication]
protocol = xmlrpc/ssl
#if $self.metadata.uuid != None
user = $self.metadata.uuid
#endif
#if $self.metadata.password != None
password = $self.metadata.password
#else
password = my-password-foobat
#endif
fingerprint = d8b7423da5d8ccd0f3db29742fc8eed00b9d0848

[components]
bcfg2 = https://localhost:6789
```

In this setup, this will cause any clients that have uuids established to be set to use them in bcfg2.conf. It will also cause any clients with passwords set to use them instead of the global password. The fingerprint needs to be manually set, per−server, using the output of "bcfg2−admin fingerprint".

How Authentication Works

1. First, the client is associated with a client record. If the client specifies a uuid, it uses this instead of the results of a dns or address lookup.
2. Next, the ip address is verified against the client record. If the address doesn't match, then the client must be set to location=floating
3. Finally, the password is verified. If the client is set to secure mode, the only its per−client password is accepted. If it is not set to secure mode, then either the global password or per−client password will be accepted

Failure during any of these stages results in authentication failure. Note that clients set into secure mode that do not have per−client passwords set will not be able to connect.